UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/837,671	04/18/2001	David L. Detlefs	004-5723	9850

22120 7590 03/04/2005

ZAGORIN O'BRIEN GRAHAM LLP
7600B N. CAPITAL OF TEXAS HWY.
SUITE 350
AUSTIN, TX 78731

EXAMINER	
BULLOCK JR, LEWIS ALEXANDER	

ART UNIT	PAPER NUMBER
2127	

DATE MAILED: 03/04/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/837,671	**Applicant(s)** DETLEFS ET AL.	
	Examiner Lewis A. Bullock, Jr.	**Art Unit** 2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 February 2005*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-35* is/are pending in the application.

4a) Of the above claim(s) *22-28* is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,7-10,13,14,16 and 29-35* is/are rejected.

7)☒ Claim(s) *4-6,11,12,15 and 17-21* is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *18 April 2001* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 112

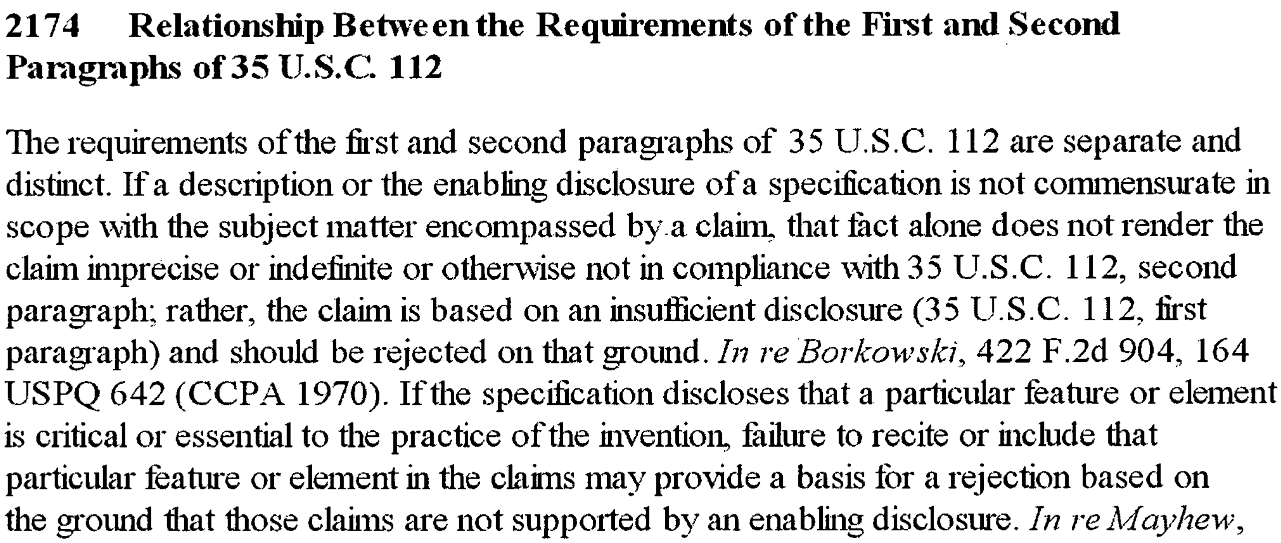
1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

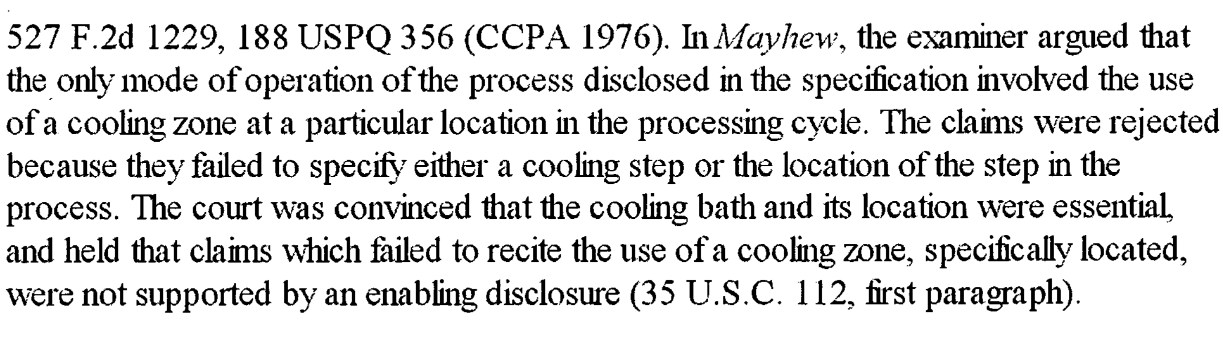
> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

M.P.E.P. 2174 states:



2174 Relationship Between the Requirements of the First and Second Paragraphs of 35 U.S.C. 112

The requirements of the first and second paragraphs of 35 U.S.C. 112 are separate and distinct. If a description or the enabling disclosure of a specification is not commensurate in scope with the subject matter encompassed by a claim, that fact alone does not render the claim imprecise or indefinite or otherwise not in compliance with 35 U.S.C. 112, second paragraph; rather, the claim is based on an insufficient disclosure (35 U.S.C. 112, first paragraph) and should be rejected on that ground. *In re Borkowski*, 422 F.2d 904, 164 USPQ 642 (CCPA 1970). If the specification discloses that a particular feature or element is critical or essential to the practice of the invention, failure to recite or include that particular feature or element in the claims may provide a basis for a rejection based on the ground that those claims are not supported by an enabling disclosure. *In re Mayhew*,



527 F.2d 1229, 188 USPQ 356 (CCPA 1976). In *Mayhew*, the examiner argued that the only mode of operation of the process disclosed in the specification involved the use of a cooling zone at a particular location in the processing cycle. The claims were rejected because they failed to specify either a cooling step or the location of the step in the process. The court was convinced that the cooling bath and its location were essential, and held that claims which failed to recite the use of a cooling zone, specifically located, were not supported by an enabling disclosure (35 U.S.C. 112, first paragraph).

3. Claims 1-3, 7-10, 13, 14, 16 and 29-35 are rejected under 35 U.S.C. 112, first paragraph. The lock-free pointer operations using a synchronization primitive is critical or essential to the practice of the invention, but not included in the claim(s) is not enabled by the disclosure. See *In re Mayhew*, 527 F.2d 1229, 188 USPQ 356 (CCPA 1976). Applicant has disclosed in the responses filed 7/21/04 and 2/2/05 that the cited invention is a working model that does not allow for premature reclaiming of a shared object which is directly opposite to the prior art teachings. A review of the specification clearly states that this is achieved in the cited technique using a synchronization primitive, such as a double compare and swap (DCAS) operation, to update the reference count of the object atomically with a check that a pointer to the object still exists. This cited limitation is critical to Applicant's invention in order to avoid premature reclaimation of an object. See specification, pg 7, paragraph 1021; pg. 9, paragraph 1026; pg. 10, paragraph 1029; pg. 23, paragraph 1050). All of the cited claims above do not illustrate the pointer operations using a synchronization primitive to increment and decrement reference counters. Therefore, these claims will fall victim to the same problems that Applicant's invention is attempting to alleviate, i.e. that a reference count can potentially have the wrong count value such that a garbage collector de-allocates the object even though a reference is still exist. Claim 1's detailing accessing pointers to the shared objects using lock-free pointer operations to coordinate modification of respective counts does not illustrate the critical and essential operation to alleviate premature reclaimation of the object. Claim 29's detailing of employing one or more lock-free pointer operations to maintain reference counts for one or more accessed

component objects thereof does not illustrate the critical and essential operation to alleviate premature reclaimation of the object. Claim 34's detailing of means for coordinating competing access to the shared object using one or more reference counts and pointer manipulations also does not illustrate the critical and essential operation to alleviate premature reclaimation of the object. All of the cited claims allow for premature reclaimation because the reclaimation is based on the reference count value. Applicant's essential limitation that the lock-free pointer operations use a synchronization primitive for incrementing and decrementing / modifying the reference counter is critical to ensure that the system does not prematurely reclaimed the shared object by having an incorrect reference counter.

4. Claims 1-3, 7-10, 13, 14, 16 and 29-35 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential elements, such omission amounting to a gap between the elements. See MPEP § 2172.01. The omitted elements are: The lock-free pointer operations using a synchronization primitive to modify the reference counter is critical or essential to the practice of the invention, but not included in the claim(s) is not enabled by the disclosure. See *In re Mayhew*, 527 F.2d 1229, 188 USPQ 356 (CCPA 1976). Applicant has disclosed in the responses filed 7/21/04 and 2/2/05 that the cited invention is a working model that does not allow for premature reclaiming of a shared object which is directly opposite the prior art teachings. A review of the specification clearly states that this is achieved in the cited technique using a synchronization primitive, such as a double compare and swap (DCAS) operation, to update the reference count of the object atomically with a check

that a pointer to the object still exists. This cited limitation is critical to Applicant's invention in order to avoid premature reclaimation of an object. See specification, pg 7, paragraph 1021; pg. 9, paragraph 1026; pg. 10, paragraph 1029; pg. 23, paragraph 1050). All of the cited claims above do not illustrate the pointer operations using a synchronization primitive to increment and decrement reference counters. Therefore, these claims will fall victim to the same problems that Applicant's invention is attempting to alleviate, i.e. that a reference count can potentially have the wrong count value such that a garbage collector de-allocates the object even though a reference is still exist. Claim 1's detailing accessing pointers to the shared objects using lock-free pointer operations to coordinate modification of respective counts does not illustrate the critical and essential operation to alleviate premature reclaimation of the object. Claim 29's detailing of employing one or more lock-free pointer operations to maintain reference counts for one or more accessed component objects thereof does not illustrate the critical and essential operation to alleviate premature reclaimation of the object. Claim 34's detailing of means for coordinating competing access to the shared object using one or more reference counts and pointer manipulations also does not illustrate the critical and essential operation to alleviate premature reclaimation of the object. All of the cited claims allow for premature reclaimation because the reclaimation is based on the reference count value. Applicant's essential limitation that the lock-free pointer operations use a synchronization primitive for incrementing and decrementing / modifying the reference counter is critical to ensure that the system does not prematurely reclaimed the shared object by having an incorrect reference counter.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1, 2, 7, 9, 29, 30 and 33-35 are rejected under 35 U.S.C. 102(b) as being anticipated by "Managing Long Linked Lists Using Lock-Free Techniques" by FAROOK.

As to claim 1, FAROOK teaches a method of providing storage reclamation in a multiprocessor computer system (a parallel environment in which each process executes alone on a processor) (pg. 15, Performance of the New Algorithm), the method comprising: maintaining respective reference counts (counter fields) for shared objects (list nodes) (pg. 9, "Each node in the linked list consists of four fields; a data field, two pointer fields, and a count field…the counter field in each list node is used…"); accessing pointers to the shared objects (list nodes) using lock-free pointer operations (prev->traverseptr-> counter++ operation / prev->counter- - within the lockfree algorithm cursor) to coordinate modification of respective reference counts (increment reference counter / decrement reference counter) (pg. 10); free storage associated with a particular one of the shared objects (list nodes) only once the corresponding reference count (counter fields) indicates that the particular shared object is unreferenced (pg. 12, "The code then decrements the counter of node 'prev' (as it is no longer being used) and, if possible releases the deleted node (line 3) to reclaim space."; pg. 9, "A process attempting a node deletion must first verify that its counter field is zero before proceeding.").

As to claim 2, FAROOK teaches the lock-free pointer operations (prev->traverseptr-> counter++ operation / prev->counter- - within the lock-free algorithm cursor) ensure that: if a number of pointers referencing the particular shared object (list nodes) is non-zero, then so too is the corresponding reference count (via as the user moves from node to node it increments the counter field of the node its visiting and decrements it when it is leaving) (pg. 9, 2nd paragraph); and if no pointers reference the particular shared object (list nodes), then the corresponding reference count eventually becomes zero (via the reference count is not increment since no node is visiting it) (pg. 9, 2nd paragraph).

As to claim 7, FAROOK teaches the pointer operations include a destroy operation (prev->counter- -) that: decrements a reference count (counter field) of a shared object (list node) identified by a supplied pointer value (key); and frees the identified shared object if the corresponding reference count has reached zero (via release instruction) (see fig. 6 and 7, pgs.11-12).

As to claim 9, FAROOK teaches employed in access operations (prev->traverseptr-> counter++ operation / prev->counter- -) on a composite shared object (linked list) that includes zero or more of the shared objects (list nodes) (pg. 8-13).

As to claim 29, FAROOK teaches a computer program product (computer instructions) encoded in at least one computer readable medium comprising (shared memory machine) (abstract; pg. 9, "Each node in the linked list consists of four fields; a data field, two pointer fields, and a count field…the counter field in each list node is used…"): a representation of a shared object (linked-list) that is instantiable as zero or more component objects (list nodes) in dynamically allocated shared storage (shared memory machine) of a multiprocessor (a parallel environment in which each process executes alone on a processor) (pg. 15, Performance of the New Algorithm); at least one instruction sequence executable by respective processors of the multiprocessor, the at least one instruction sequence implementing at least one access operation (cursor / insert / delete) on the shared object (linked-list) and employing one or more lock-free pointer operations (prev->traverseptr-> counter++ operation / prev->counter- - within the lock-free algorithm cursor) to maintain reference counts (counter fields) for one or more accessed component objects (list nodes) thereof (increment reference counter / decrement reference counter) (pg. 10; see figs. 6-9; pg. 9, section 4.2 Linked List Traversal – pg. 13, "Otherwise, the 'prev' node's counter field is decremented (line 14) and the appropriate result status is returned."; pg. 12, "The code then decrements the counter of node 'prev' (as it is no longer being used) and, if possible releases the deleted node (line 3) to reclaim space."; pg. 9, "A process attempting a node deletion must first verify that its counter field is zero before proceeding."; see also pg. 14-15); and the at least one instruction sequence further implementing explicit reclamation of the component objects (list nodes), thereby freeing storage associated with a particular one of the component objects (list nodes) only once the corresponding reference count (counter fields) indicates that the particular component object is unreferenced (via the instruction that if target counter or prev counter is equal to 0 then to release that node) (see fig. 7, pg. 12).

As to claim 30, FAROOK teaches the zero or more component objects (list nodes) of the shared object (linked list) are organized as a linked-list; and wherein the at least one access operation (cursor / insert / delete) supports concurrent access to the linked-list (non-blocking concurrent access) (pg. 9; abstract; pg. 14-15, "We begin by noting that non-adjacent updates do not affect one another since, synchronization is localized (i.e. fine granularity concurrency control).").

As to claim 33, FAROOK teaches the computer readable medium is electronic storage medium (shared memory machine) (abstract).

As to claim 34, FAROOK teaches an apparatus (system) comprising: plural processors (a parallel environment in which each process executes alone on a processor) (pg. 15, Performance of the New Algorithm); a store addressable by the plural processors (shared memory storing the linked list); one or more shared pointer variables (head / tail pointers) accessible by each of the plural processors for referencing a shared object (linked list) encoded in the store; means for coordinating competing access to the shared object (linked list / list node) using one or more reference counts (counter fields) and pointer manipulations (via instructions manipulating the next or prev pointer values in the cursor function / delete function / insert function) that employ one or more lock-free pointer operations (prev->traverseptr-> counter++ operation / prev->counter- - within the lock-free algorithm cursor) to ensure that if the number of pointers to the shared object (linked list / list node) is non-zero (node is being referenced / visited) particular, then so too is the corresponding reference count (via each process increments the counter of a node it is about to traverse) and further that if no pointers reference the shared object (process is leaving node), then the corresponding reference count (counter field) eventually becomes zero (via each process decrements the counter of a node it is leaving) (pgs 8-13, in particular pg. 10).

As to claim 35, FAROOK teaches means for freeing the shared object (list node) only once the corresponding reference count (counter field) indicates that the shared object is unreferenced (via the instruction that if target counter or prev counter is equal to 0 then to release that node) (see fig. 7, pg. 12).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the

> invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over "Managing Long Linked Lists Using Lock-Free Techniques" by FAROOK.

As to claim 3, FAROOK teaches that as concurrent processes access a shared object, i.e. visit a shared node of the shared object, they increment that nodes reference count (abstract, pg. 8-15). It is obvious to one skilled in the art that since accesses to the shared object are concurrently that when both processes access the object at the same time neither would have the true reference count, i.e. if the reference count of a shared object is one and concurrent access is made by concurrently accessing processes each process would increment to two wherein the correct amount of pointers is three.

7. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over "Managing Long Linked Lists Using Lock Free Techniques" by FAROOK in view of "Simple, Fast, and Practical Non-Blocking and Blocking Concurrent Queue Algorithms" by MICHAEL.

As to claim 10, FAROOK teaches the composite shared object is a linked list and the shared objects are nodes of the linked list and wherein the access operations implement push (insert) and pop (delete) accesses to opposing ends of the linked list (pg. 8-13). However, FAROOK does not teach the linked list is a doubled ended queue.

MICHAEL teaches a queue implemented as a linked list having a head and tail pointers wherein nodes are pushed (enqueue operation) after the last node in the linked list and pop (dequeue operation) from the beginning of the linked list (pg. 3-5, Sections Algorithms (2) and Correctness (3)). It would be obvious that based on the combination that FAROOK's linked list is a queue capable of being manipulated as disclosed in MICHAEL. Therefore, it would be obvious to one skilled in the art to combine the teachings of FAROOK with the teachings of MICHAEL in order to facilitate access operations of a non-blocking concurrent queue to proceed concurrently (abstract).

8. Claims 31 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Managing Long Linked Lists Using Lock-Free Techniques" by FAROOK in view of "Garbage Collection: Algorithms for Automatic Dynamic Memory Management" by JONES.

As to claim 31, FAROOK teaches component objects (list nodes) having a reference count (counter field) (pg. 8). However, FAROOK does not teach using a mutator to provide explicit reclaimation.

JONES teaches partially implementing a mutator (mutator) that provides explicit reclamation of the dynamically allocated shared storage (concurrent reference counter) (via notifying the collector) (pg. 200-201). Therefore, it would be obvious to combine the teachings of FAROOK with the teachings of JONES in order to efficiently manipulate reference counters in a concurrent environment (pg. .200).

As to claim 32, JONES teaches partially implementing a garbage collector (garbage collector) that reclaims shared storage dynamically-allocated from a mutator (mutator) (pg. 200-201).

Allowable Subject Matter

9. Claims 4-6, 11, 12, 15, and 17-21 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

10. The following is a statement of reasons for the indication of allowable subject matter: The cited claims contain allowable subject matter for at least the following reasoning: All of the claims detail a plurality of lock-free pointer operations, i.e a load operation, a store operation, a copy operation, and a push access operation that are not taught by any of the prior art of record. Each of the cited operations perform a plurality of compare-and-swap and double-compare-and swap primitives that affect not only the reference count but also the pre-load value of the local pointer value, the pre-store value of the shared pointer variable, the pre-copy value of the local pointer value, or the in pre-splice pointer values. By performing DCAS and CAS operations as disclosed in the claims, the invention allows for the separation of the updates of reference counts from the updates of the pointers themselves (pg. 7, par. 1021). All of the prior art of record detail common practices of allowing the pointers to record the exact number of pointers to an object. FAROOK teaches this common practice in that each insert or delete operation manipulates the counter for the accessed node however this

manipulation is not performed by the DCAS or CAS operations with the access operations. By performing the pointer operations using a synchronization primitive, the invention avoids the premature reclaiming of shared objects. Therefore, the claims are allowable over the prior art of record.

Response to Arguments

11. Applicant's arguments filed 2/2/05 have been fully considered but they are not persuasive. Applicant argued that the Farook allows for premature reclamation which is avoided by Applicant's claimed invention. In response, the examiner points to the arguments for the 35 U.S.C. 112 rejections made above. In addition, the examiner has defined lock-free pointer operations to be operations that use pointers to perform an operation. The operations can receive these pointers by value or they are part of the operation. Farook teaches that the cursor algorithm has a pointer operations, prev->traverseptr-> counter++ instruction and prev->counter- -. These instructions are pointer operations that manipulate or coordinate modification of a reference counter, since prev is an reference to one of the shared objects. Since the overall algorithm is non-blocking and can be called numerous times, instructions of the algorithm are lock-free. Therefore, Farook teaches the instructions as detailed in the claims as currently written.

In regards to claims 8 and 13, this rejection is moot in view of the prior art rejection to claims 8 and 13 being withdrawn. However, claims 8 and 13 are rejected under 35 U.S.C. 112 as detailed above.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lewis A. Bullock, Jr. whose telephone number is (571) 272-3759. The examiner can normally be reached on Monday-Friday, 8:30 am - 5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

LEWIS A. BULLOCK, JR.
PRIMARY EXAMINER

March 1, 2005